

NAME

locate – list files in databases that match a pattern

SYNOPSIS

locate [-d *path* | --database=*path*] [-e | -E | --[non-]existing] [-i | --ignore-case] [-0 | --null] [-c | --count] [-w | --wholename] [-b | --basename] [-l *N* | --limit=*N*] [-S | --statistics] [-r | --regex] [--regextype *R*] [--max-database-age *D*] [-P | -H | --nofollow] [-L | --follow] [--version] [-A | --all] [-p | --print] [--help] *pattern...*

DESCRIPTION

This manual page documents the GNU version of **locate**. For each given pattern, **locate** searches one or more databases of file names and displays the file names that contain the pattern. Patterns can contain shell-style metacharacters: `*`, `?`, and `[]`. The metacharacters do not treat `'` or `.` specially. Therefore, a pattern `foo*bar` can match a file name that contains `foo3/bar`, and a pattern `*duck*` can match a file name that contains `lake/.ducky`. Patterns that contain metacharacters should be quoted to protect them from expansion by the shell.

If a pattern is a plain string — it contains no metacharacters — **locate** displays all file names in the database that contain that string anywhere. If a pattern does contain metacharacters, **locate** only displays file names that match the pattern exactly. As a result, patterns that contain metacharacters should usually begin with a `*`, and will most often end with one as well. The exceptions are patterns that are intended to explicitly match the beginning or end of a file name.

The file name databases contain lists of files that were on the system when the databases were last updated. The system administrator can choose the file name of the default database, the frequency with which the databases are updated, and the directories for which they contain entries; see **updatedb**(1).

If **locate**'s output is going to a terminal, unusual characters in the output are escaped in the same way as for the `--print` action of the **find** command. If the output is not going to a terminal, file names are printed exactly as-is.

OPTIONS

`-0`, `--null`

Use ASCII NUL as a separator, instead of newline.

`-A`, `--all`

Print only names which match all non-option arguments, not those matching one or more non-option arguments.

`-b`, `--basename`

Results are considered to match if the pattern specified matches the final component of the name of a file as listed in the database. This final component is usually referred to as the 'base name'.

`-c`, `--count`

Instead of printing the matched filenames, just print the total number of matches we found, unless `--print` (`-p`) is also present.

`-d path`, `--database=path`

Instead of searching the default file name database, search the file name databases in *path*, which is a colon-separated list of database file names. You can also use the environment variable **LOCATE_PATH** to set the list of database files to search. The option overrides the environment variable if both are used. Empty elements in the path are taken to be synonyms for the file name of the default database. A database can be supplied on stdin, using `-` as an element of *path*. If more than one element of *path* is `-`, later instances are ignored (and a warning message is printed).

The file name database format changed starting with GNU **find** and **locate** version 4.0 to allow machines with different byte orderings to share the databases. This version of **locate** can automatically recognize and read databases produced for older versions of GNU **locate** or Unix versions of **locate** or **find**. Support for the old locate database format will be discontinued in a future release.

- e, --existing*
Only print out such names that currently exist (instead of such names that existed when the database was created). Note that this may slow down the program a lot, if there are many matches in the database. If you are using this option within a program, please note that it is possible for the file to be deleted after **locate** has checked that it exists, but before you use it.
- E, --non-existing*
Only print out such names that currently do not exist (instead of such names that existed when the database was created). Note that this may slow down the program a lot, if there are many matches in the database.
- help* Print a summary of the options to **locate** and exit.
- i, --ignore-case*
Ignore case distinctions in both the pattern and the file names.
- l N, --limit=N*
Limit the number of matches to N. If a limit is set via this option, the number of results printed for the *-c* option will never be larger than this number.
- L, --follow*
If testing for the existence of files (with the *-e* or *-E* options), consider broken symbolic links to be non-existing. This is the default.
- max-database-age D*
Normally, **locate** will issue a warning message when it searches a database which is more than 8 days old. This option changes that value to something other than 8. The effect of specifying a negative value is undefined.
- m, --mmap*
Accepted but does nothing, for compatibility with BSD **locate**.
- P, -H, --nofollow*
If testing for the existence of files (with the *-e* or *-E* options), treat broken symbolic links as if they were existing files. The *-H* form of this option is provided purely for similarity with **find**; the use of *-P* is recommended over *-H*.
- p, --print*
Print search results when they normally would not, because of the presence of *--statistics* (*-S*) or *--count* (*-c*).
- r, --regex*
The pattern specified on the command line is understood to be a regular expression, as opposed to a glob pattern. The Regular expressions work in the same way as in **emacs** except for the fact that "." will match a newline. GNU **find** uses the same regular expressions. Filenames whose full paths match the specified regular expression are printed (or, in the case of the *-c* option, counted). If you wish to anchor your regular expression at the ends of the full path name, then as is usual with regular expressions, you should use the characters *^* and *\$* to signify this.
- regextype R*
Use regular expression dialect R. Supported dialects include 'findutils-default', 'posix-awk', 'posix-basic', 'posix-egrep', 'posix-extended', 'posix-minimal-basic', 'awk', 'ed', 'egrep', 'emacs', 'gnu-awk', 'grep' and 'sed'. See the Texinfo documentation for a detailed explanation of these dialects.
- s, --stdio*
Accepted but does nothing, for compatibility with BSD **locate**.
- S, --statistics*
Print various statistics about each locate database and then exit without performing a search, unless non-option arguments are given. For compatibility with BSD, *-S* is accepted as a synonym for *--statistics*. However, the output of **locate -S** is different for the GNU and BSD

implementations of **locate**.

--version

Print the version number of **locate** and exit.

-w, --wholename

Match against the whole name of the file as listed in the database. This is the default.

ENVIRONMENT

LOCATE_PATH

Colon-separated list of databases to search. If the value has a leading or trailing colon, or has two colons in a row, you may get results that vary between different versions of **locate**.

HISTORY

The **locate** program started life as the BSD fast find program, contributed to BSD by James A. Woods. This was described by his paper *Finding Files Fast* which was published in *Usenix ;login:*, Vol 8, No 1, February/March, 1983, pp. 8-10. When the **find** program began to assume a default **-print** action if no action was specified, this changed the interpretation of **find pattern**. The BSD developers therefore moved the fast find functionality into **locate**. The GNU implementation of **locate** appears to be derived from the same code.

Significant changes to **locate** in reverse order:

- 4.3.7 Byte-order independent support for old database format
- 4.3.3 **locate -i** supports multi-byte characters correctly
 - Introduced **--max_db_age**
- 4.3.2 Support for the **slocate** database format
- 4.2.22 Introduced the **--all** option
- 4.2.15 Introduced the **--regex** option
- 4.2.14 Introduced options **-L, -P, -H**
- 4.2.12 Empty items in **LOCATE_PATH** now indicate the default database
- 4.2.11 Introduced the **--statistics** option
- 4.2.4 Introduced **--count** and **--limit**
- 4.2.0 Glob characters cause matching against the whole file name
- 4.0 Introduced the **LOCATE02** database format
- 3.7 **Locate** can search multiple databases

BUGS

The **locate** database correctly handles filenames containing newlines, but only if the system's sort command has a working **-z** option. If you suspect that **locate** may need to return filenames containing newlines, consider using its **--null** option.

REPORTING BUGS

GNU findutils online help: <<https://www.gnu.org/software/findutils/#get-help>>

Report any translation bugs to <<https://translationproject.org/team/>>

Report any other issue via the form at the GNU Savannah bug tracker:

<<https://savannah.gnu.org/bugs/?group=findutils>>

General topics about the GNU findutils package are discussed at the *bug-findutils* mailing list:

<<https://lists.gnu.org/mailman/listinfo/bug-findutils>>

COPYRIGHT

Copyright © 1994–2024 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <<https://gnu.org/licenses/gpl.html>>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

find(1), **updatedb(1)**, **xargs(1)**, **glob(3)**, **locatedb(5)**

Full documentation <<https://www.gnu.org/software/findutils/locate>>
or available locally via: **info locate**